Zilog - Z8019520FSG Datasheet

Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	Z8S180
Number of Cores/Bus Width	1 Core, 8-Bit
Speed	20MHz
Co-Processors/DSP	-
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	100-QFP
Supplier Device Package	100-QFP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8019520fsg

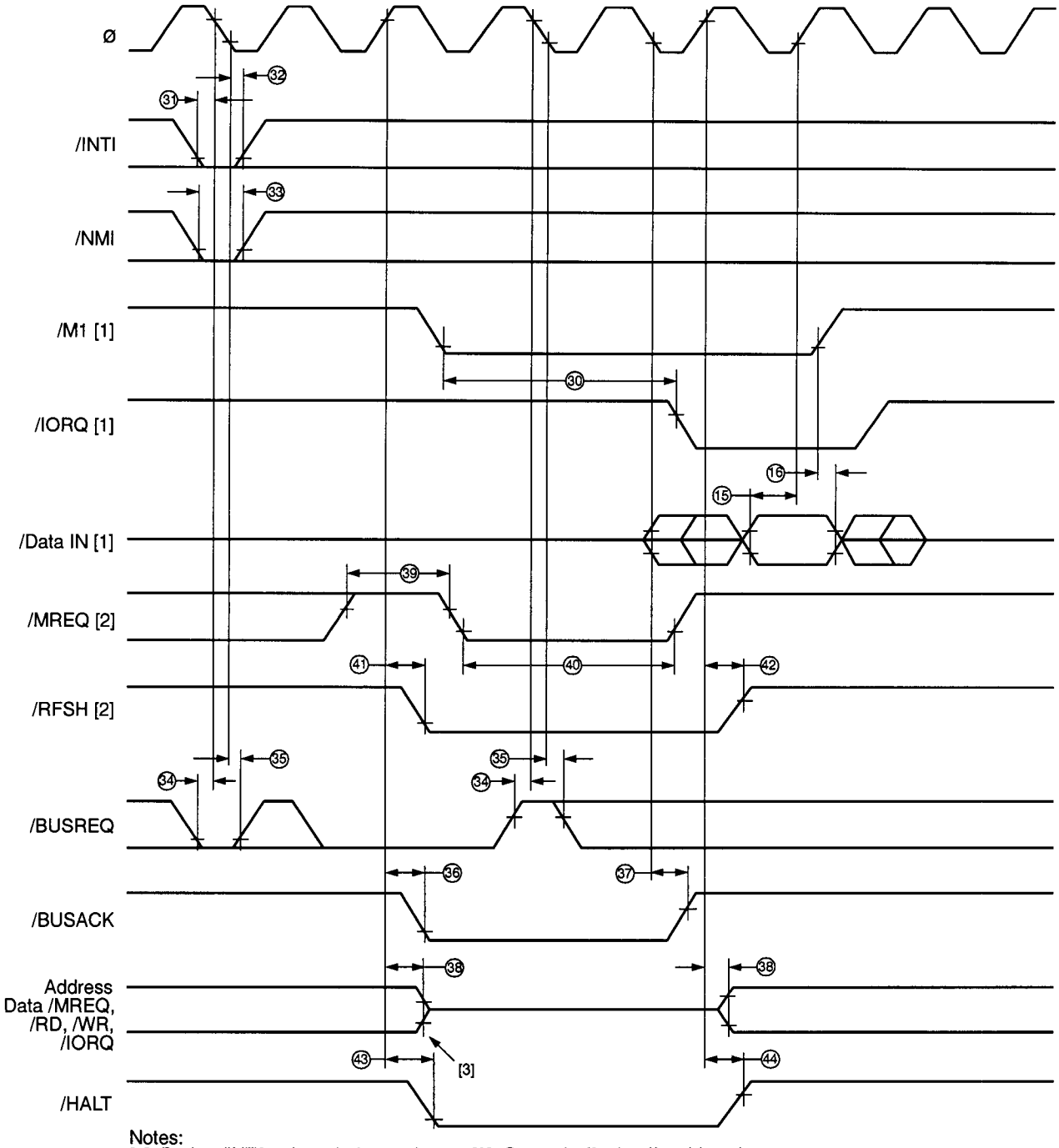


Figure 5. CPU Timing
(/INT0 Acknowledge Cycle, Refresh Cycle, BUS RELEASE mode
HALT mode, SLEEP mode, SYSTEM STOP mode)

AC CHARACTERISTICS (Continued)

Read/Write External Bus Master Timing

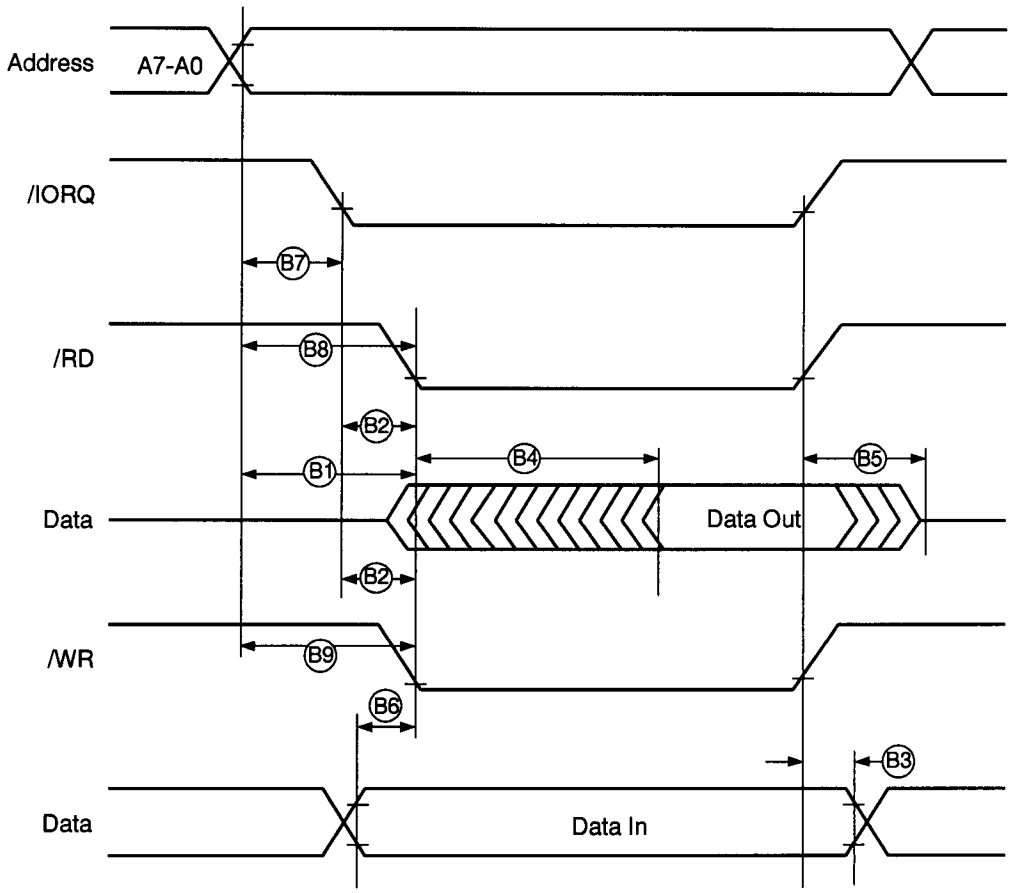


Figure 14. Read/Write External Bus Master Timing

General-Purpose I/O Timing Port Timing

Parameters referenced in Figure 15 appear in the following Tables.

Note: Port 2 timing is different, even when Bidirectional Centronics feature is not in active use.

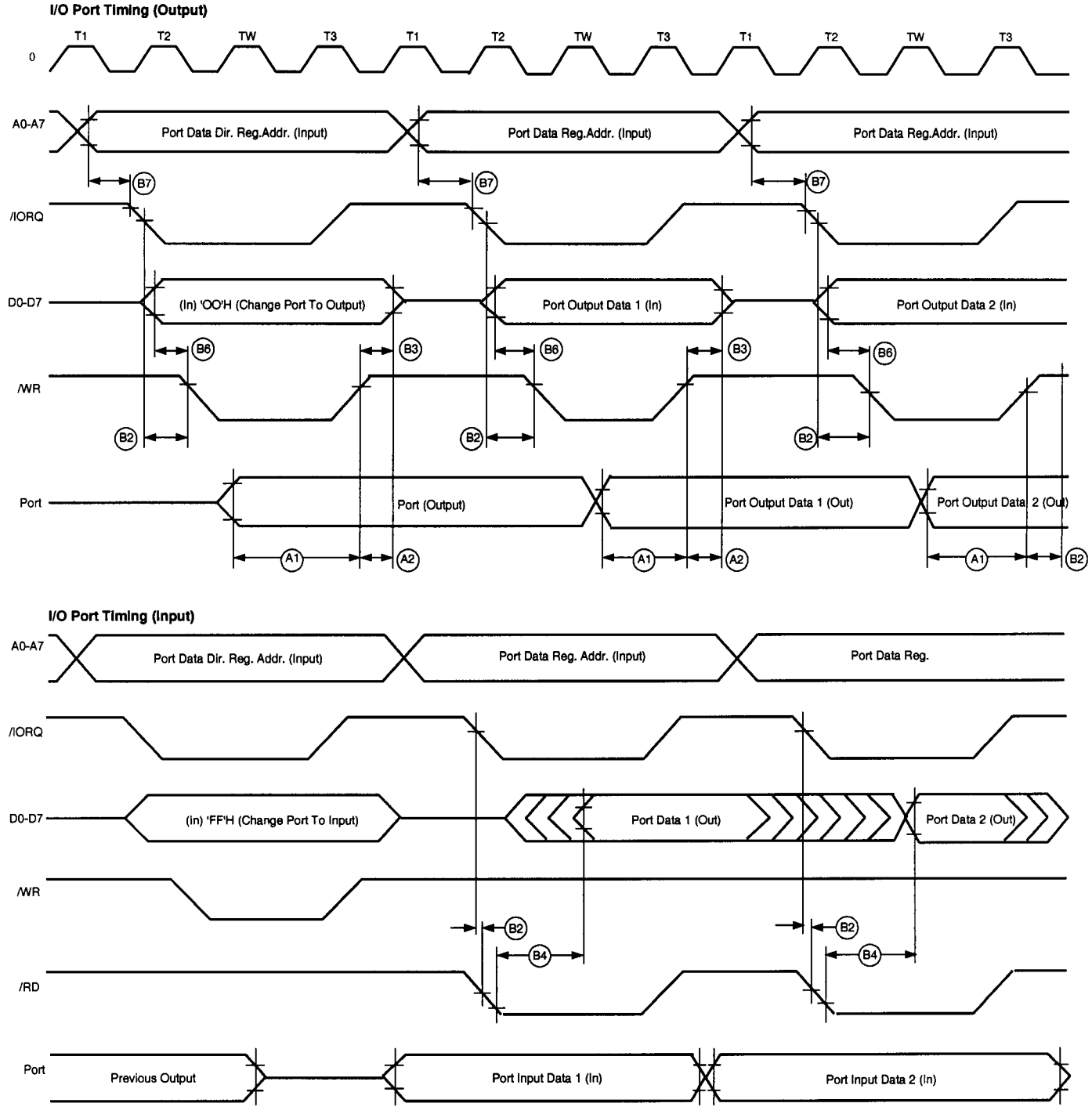


Figure 15. I/O Port Timing

EMSCC Timing

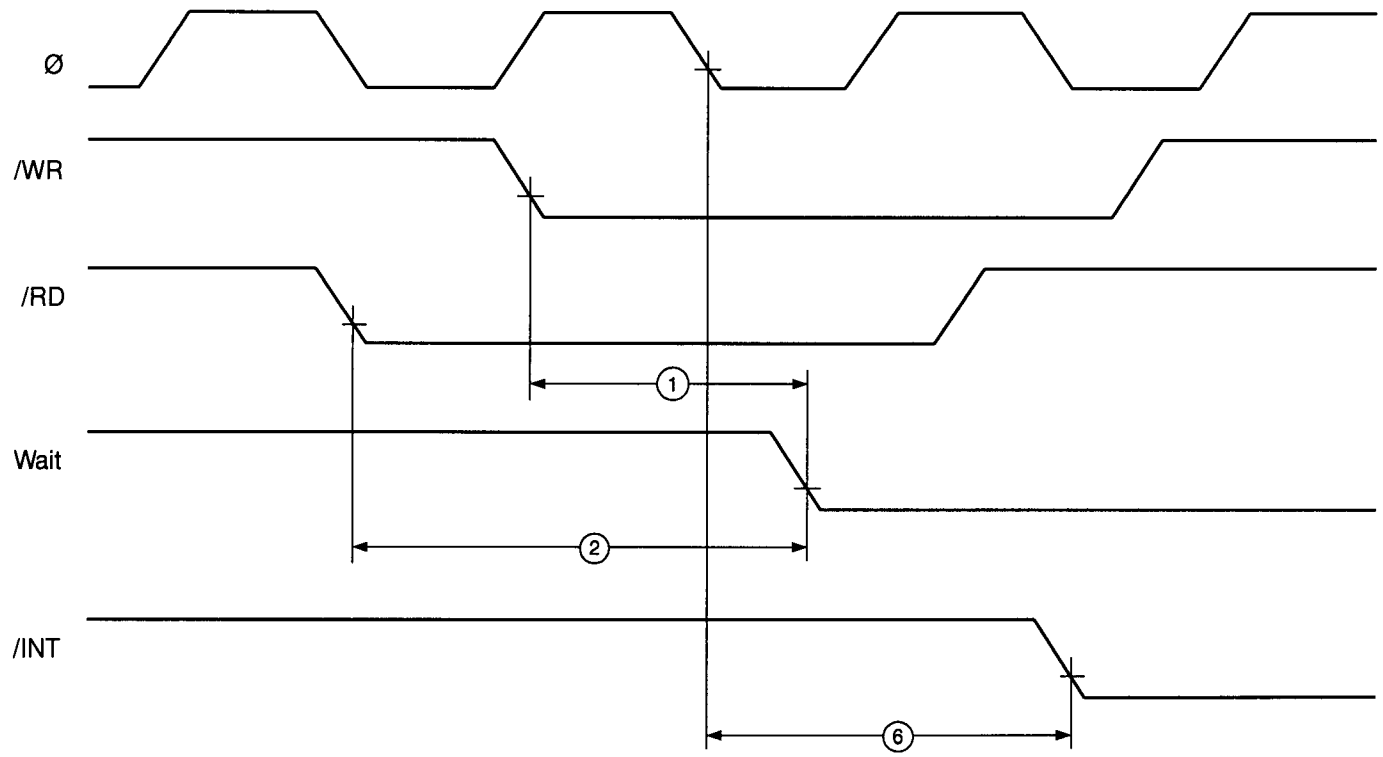


Figure 16. EMSCC AC Parameters

EMSCC Timing Parameters

No.	Symbol	Parameter	20 MHz Min	20 MHz Max	Unit
1	TdWR(W)	/WR Fall to Wait Valid Delay		50	ns
2	TdRD(W)	/RD Fall to Wait Valid Delay		50	
6	TdPC(INT)	Clock to /INT Valid Delay		160	

AC CHARACTERISTICS (Continued)

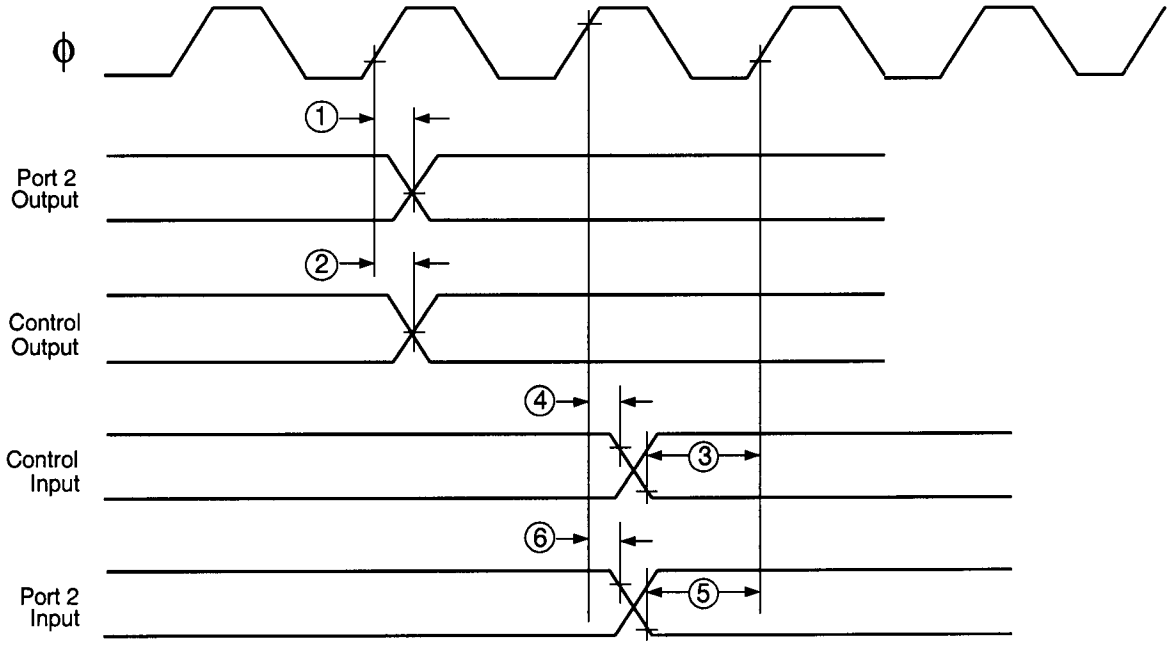


Figure 20. P1284 Bidirectional Centronics Interface Timing

P1284 Bidirectional Centronics Interface Timing

No.	Parameter	Min	Max	Units	Notes
1	CLK High to Port 2 Output		12	ns	
2	CLK High to Control Output		12	ns	(1)
3	Setup Time for Control Input to CLK High for Guaranteed Recognition	10		ns	(2)
4	Hold Time for Control Input from CLK High for Guaranteed Recognition	5		ns	(2)
5	Setup Time for Port 2 Inputs to CLK High for Guaranteed Recognition	10		ns	
6	Hold Time for Port 2 Inputs to CLK High for Guaranteed Recognition	5		ns	

Notes:

1. **Control Outputs**

Peripheral Mode	Host Mode
Busy/PtrBusy/PeriphAck	nStrobe/HostClk
nAck/PtrClk/PeriphClk	nAutoFd/HostBusy/HostAck
PError/AckDataReq/nAckReverse	nSelectIn/P1284Active
nFault/nDataAvail/nPeriphRequest	nInit/nReverseRequest
Select/Xflag	

2. **Control Inputs**

Peripheral Mode	Host Mode
Busy/PtrBusy/PeriphAck	nStrobe/HostClk
nAck/PtrClk/PeriphClk	nAutoFd/HostBusy/HostAck
PError/AckDataReq/nAckReverse	nSelectIn/P1284Active
nFault/nDataAvail/nPeriphRequest	nInit/nReverseRequest
Select/Xflag	

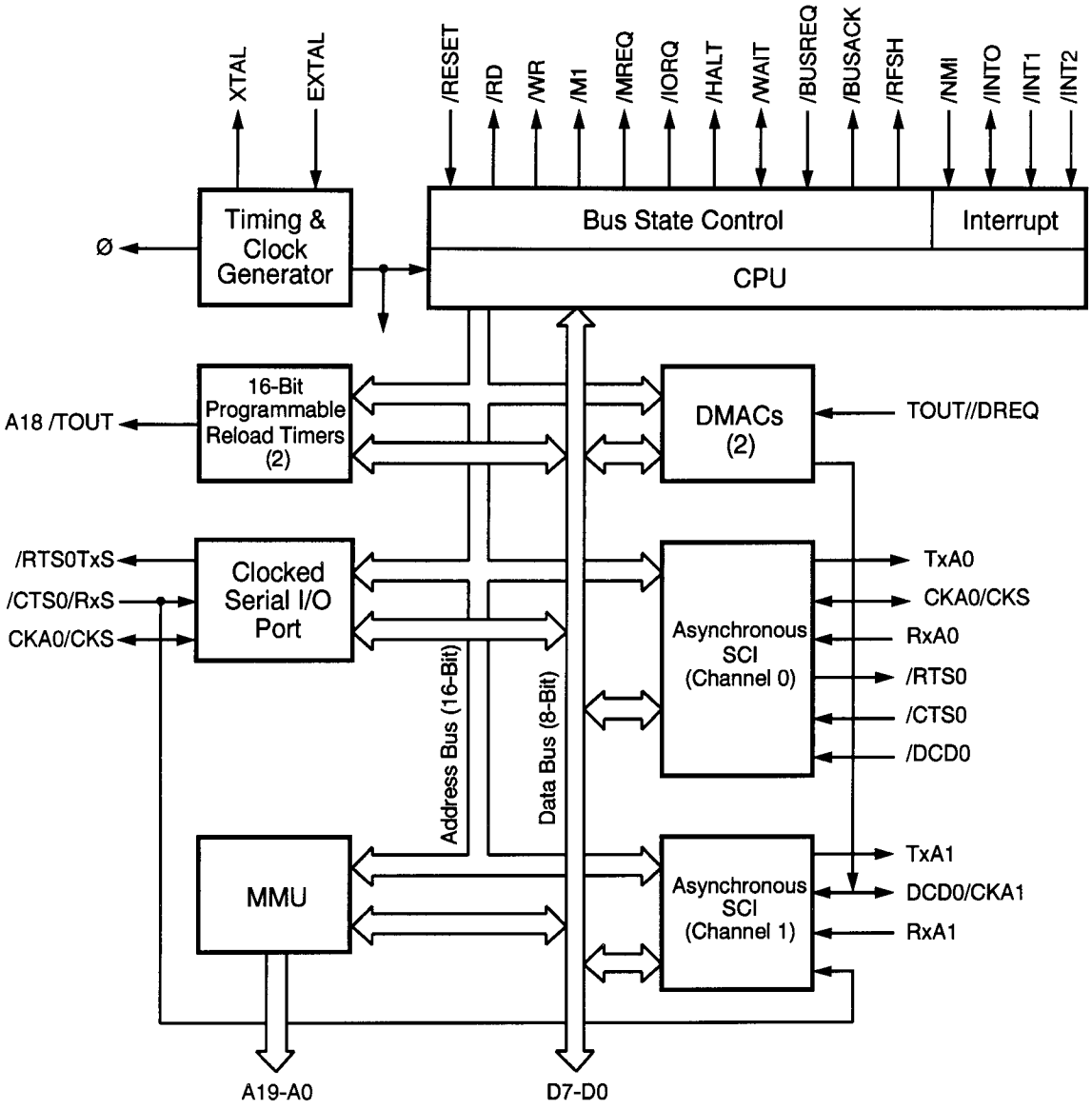


Figure 21. Z8S180 MPU Block Diagram

Z8S180 MPU REGISTERS—ASCI CHANNELS CONTROL REGISTERS

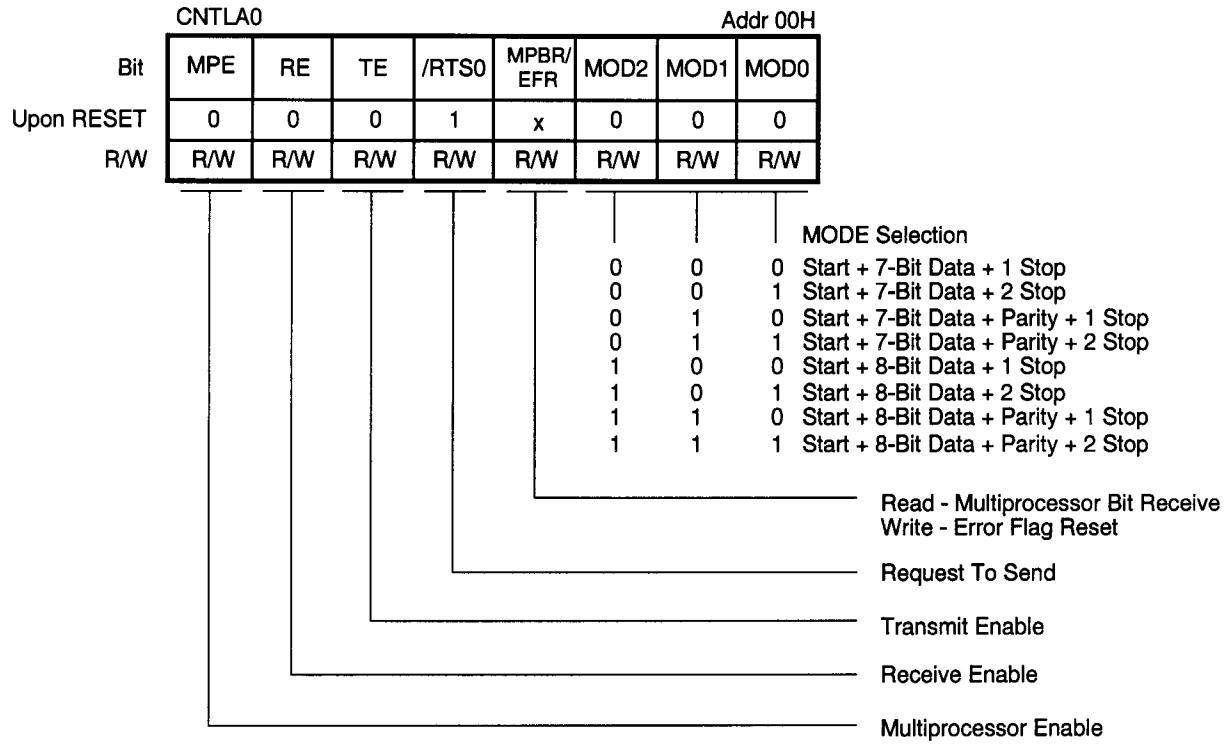


Figure 23a. ASCI Control Register A (Ch. 0)

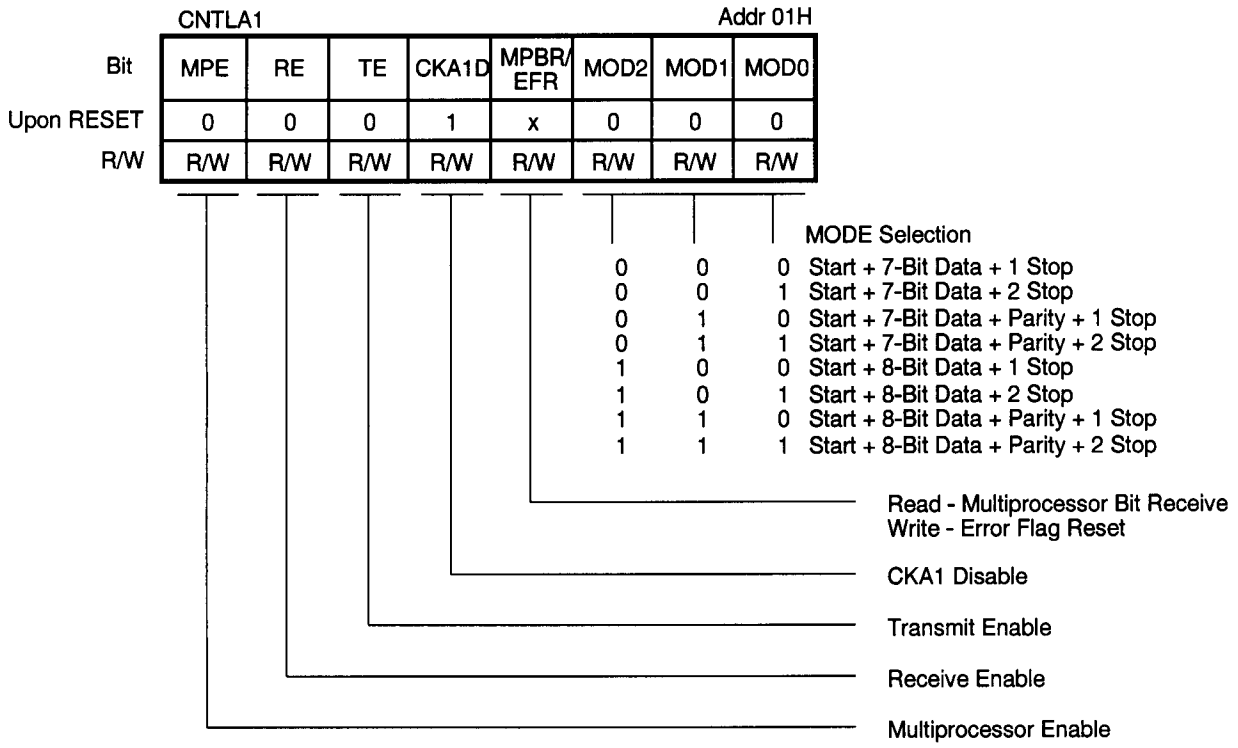
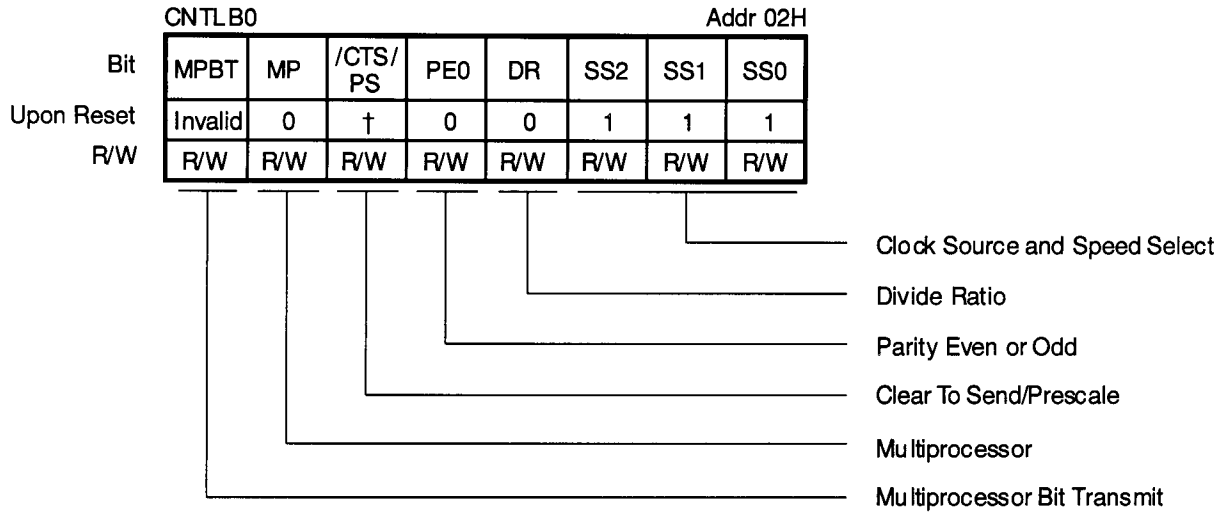


Figure 23b. ASCI Control Register A (Ch. 1)



† /CTS - Depending on the condition of /CTS pin.
PS - Cleared to 0.

General Divide Ratio	PS = 0 (Divide Ratio = 10)		PS = 1 (Divide Ratio = 30)	
SS, 2, 1, 0	**DR = 0 (x16)**	**DR = 1 (x64)**	**DR = 0 (x16)**	**DR = 1 (x64)**
000	Ø ÷ 160	Ø ÷ 640	Ø ÷ 480	Ø ÷ 1920
001	Ø ÷ 320	Ø ÷ 1280	Ø ÷ 960	Ø ÷ 3840
010	Ø ÷ 640	Ø ÷ 2560	Ø ÷ 1920	Ø ÷ 7680
011	Ø ÷ 1280	Ø ÷ 5120	Ø ÷ 3840	Ø ÷ 15360
100	Ø ÷ 2560	Ø ÷ 10240	Ø ÷ 7680	Ø ÷ 30720
101	Ø ÷ 5120	Ø ÷ 20480	Ø ÷ 15360	Ø ÷ 61440
110	Ø ÷ 10240	Ø ÷ 40960	Ø ÷ 30720	Ø ÷ 122880
111	External Clock (Frequency < Ø)			

Figure 24. ASCI Control Register B (Ch. 0)

DMA REGISTER DESCRIPTION

Bit 7. This bit should be set to 1 only when both DMA channels are set to take their requests from the same device. If this bit is 1 (it resets to 0), the channel end output of DMA channel 0 sets a flip-flop, so that thereafter the device's request is visible to channel 1, but is not visible to channel 0. The channel end output of channel 1 clears the FF, so that thereafter, the device's request is visible to channel 0, but not visible to channel 1.

Bit 6. When both DMA channels are programmed to take their requests from the same device, this bit (FF mentioned in the previous paragraph) controls which channel the device's request is presented to: 0 = DMA 0, 1 = channel 1. When bit 7 is 1, this bit is automatically toggled by the channel end output of the channels, as described above.

Bits 5-4. Reserved and should be programmed as 0.

Bits 3. This bit controls the direction and use of the TOUT/DREQ pin. When it's 0, TOUT/DREQ is the DREQ input; when it's 1, TOUT/DREQ is an output that can carry the TOUT signal from PRT1, if PRT1 is so programmed.

Bits 2-0. With "DIM1", bit 1 of DCNTL, these bits control which request is presented to DMA channel 1, as follows:

DIM1	IAR18-16	Request Routed to DMA Channel 1
0	000	ext TOUT/DREQ
0	001	ASCI0 Tx
0	010	ASCI1 Tx
0	011	EMSCC out
0	10X	Reserved, do not program.
0	1X0	Reserved, do not program.
0	111	PIA27-20 out
1	000	ext TOUT/DREQ
1	001	ASCI0 Rx
1	010	ASCI1 Rx or TOUT//DREQ pin
1	011	EMSCC in
1	10X	Reserved, do not program.
1	1X0	Reserved, do not program.
1	111	PIA27-20 in

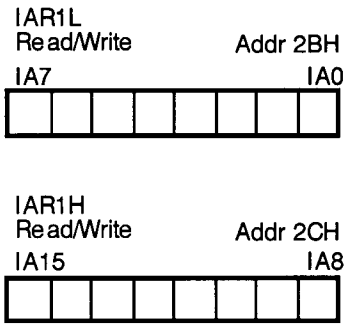


Figure 52. DMA I/O Address Registers

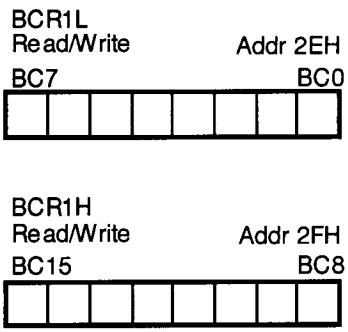


Figure 53. DMA 1 Byte Count Registers

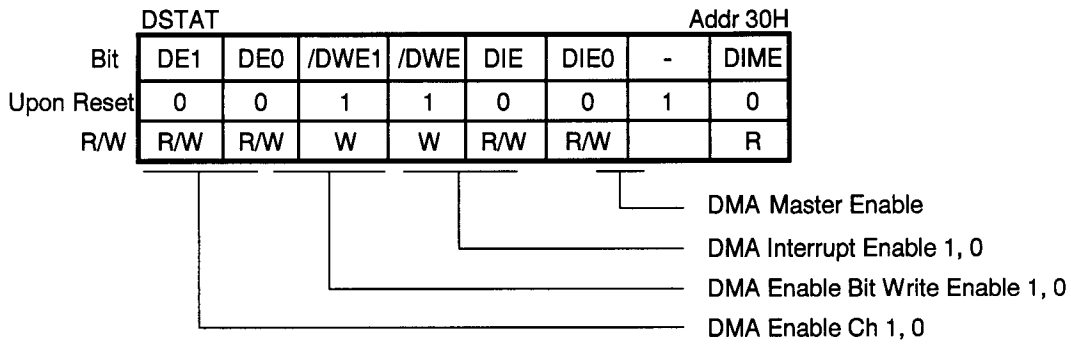


Figure 54. DMA Status Register

MMU REGISTERS

CBR — Addr 38H

Bit	CB7	CB6	CB5	CB4	CB3	CB2	CB1	CB0
Upon Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MMU Common Base Register

Figure 60. MMU Common Base Register

BBR — Addr 39H

Bit	BB7	BB6	BB5	BB4	BB3	BB2	BB1	BB0
Upon Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MMU Bank Base Register

Figure 61. MMU Bank Base Register

CBAR — Addr 3AH

Bit	CA3	CA2	CA1	CA0	BA3	BA2	BA1	BA0
Upon Reset	1	1	1	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MMU Bank Area Register

MMU Common Area Register

Figure 62. MMU Common/Bank Area Register

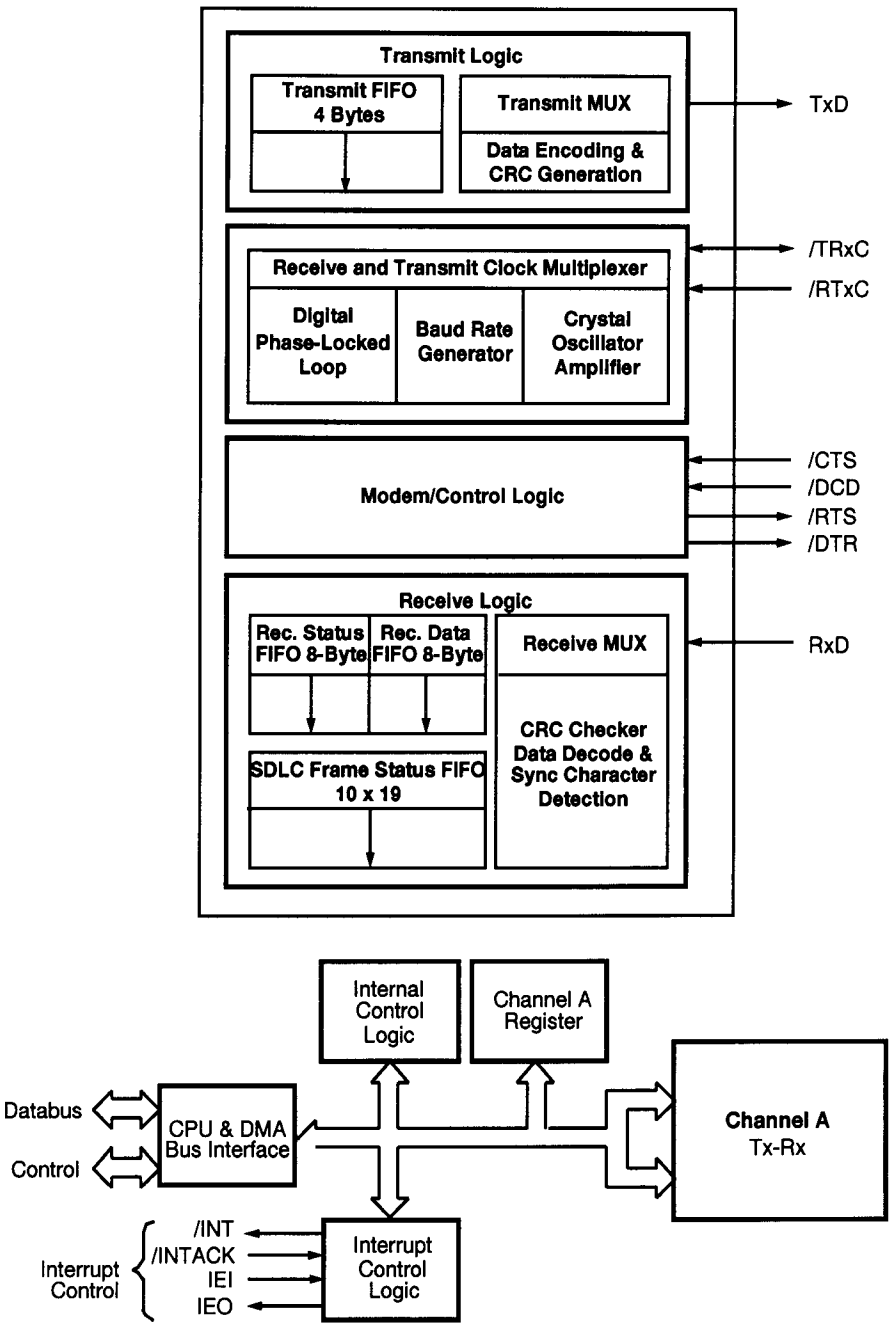


Figure 66. EMSCC Block Diagram

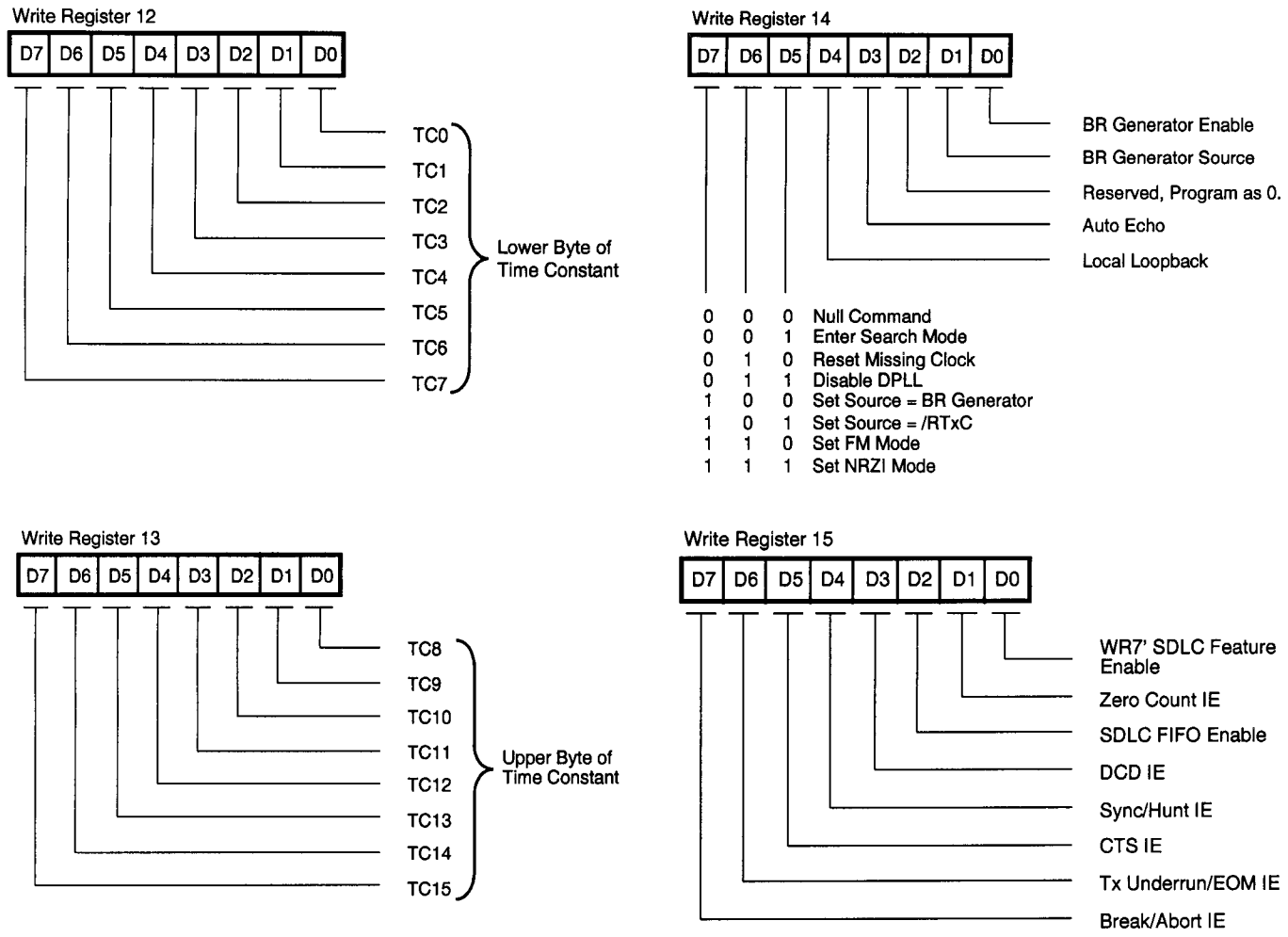


Figure 71. Write Register Bit Functions (Continued)

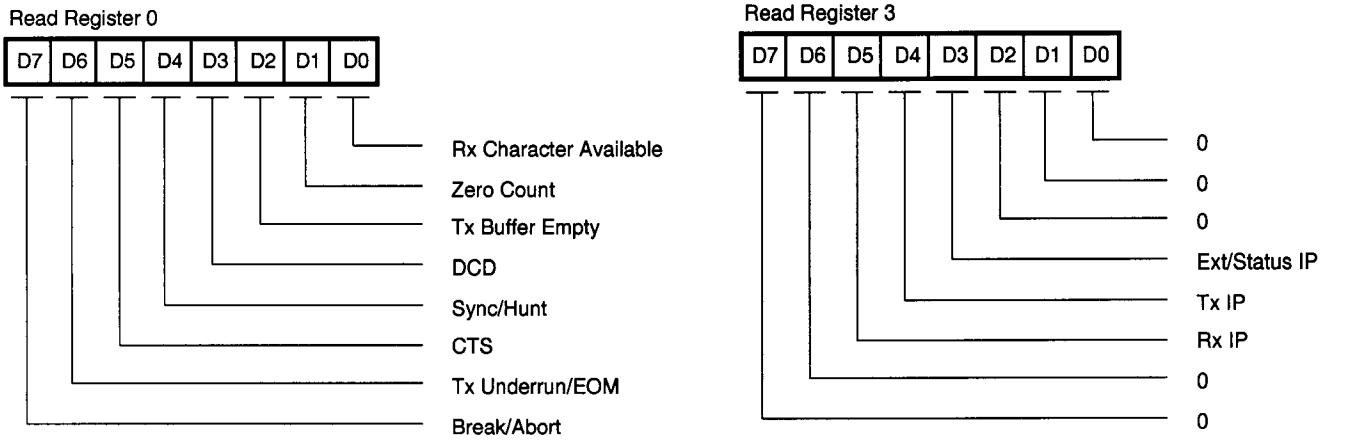


Figure 72. Read Register Bit Functions

Bidirectional Centronics Registers

Reading the Parallel Controls (PARC) register allows software to sense the state of the input signals per the current mode, plus two or three status flags:

Busy	PError	Select	nFault	nAck	IllOp	DREQ	Idle
7	6	5	4	3	2	1	0

Figure 74a. Reading PARC in a Host Mode
(I/O Address %DA)

nAutoFd	nStrobe	nSlctIn	nInit		IllOp	DREQ	Idle
7	6	5	4	3	2	1	0

Figure 74b. Reading PARC in a Peripheral Mode
(I/O Address %DA)

The controller sets IllOp (Illegal Operation) when it detects an error in the protocol, for example, if it's in Peripheral mode and it detects that the host has driven P1284Active (nSelectIn) Low at a time that mandates an immediate Abort, that is, outside one of the "windows" in which this event indicates an organized disengagement. If "status interrupts" are enabled, such an interrupt is always requested when IllOp is set. Writing PARM with NewMode=1 clears IllOp.

DREQ is the Request presented to the DMA channels, which may or may not be programmed to service this request. If not, an interrupt can be enabled when DREQ is set.

Writing to PARC allows the software to set and clear the output signals per the current mode:

1=drive nAutoFd High	1=drive nStrobe High	1=drive nSelctIn High	1=drive nInit High	1=drive nAutoFd Low	1=drive nStrobe Low	1=drive nSelctIn Low	1=drive nInit Low
7	6	5	4	3	2	1	0

Figure 75a. Writing to PARC in a Host Mode
(I/O Address %DA)

1=drive Busy High	1=drive PError High	1=drive Select High	1=drive nFault High	1=drive Busy Low	1=drive PError Low	1=drive Select Low	1=drive nFault Low
7	6	5	4	3	2	1	0

Figure 74b. Writing to PARC in a Peripheral Mode
(I/O Address %DA)

Z80185 BIDIRECTIONAL CENTRONICS P1284 CONTROLLER (Continued)

Because there are five outputs in a Peripheral mode, another register, called PARC2, allows software to change the nAck line, rather than the Select line:

1=drive Busy High	1=drive PError High	1=drive nAck High	1=drive nFault High	1=drive Busy Low	1=drive PError Low	1=drive nAck Low	1=drive nFault Low
7	6	5	4	3	2	1	0

Figure 76. Writing to PARC2 in a Peripheral Mode
(I/O Address %DB)

The Parallel mode register (PARM) includes the basic mode control of the controller:

NewMode	IdleIE	StatIE	DREQIE	Mode			
7	6	5	4	3	2	1	0

Figure 77. PARM (I/O Address %D9)

NewMode = 1 reinitializes the state machine to the initial state for the mode called out by MODE. Never change MODE without writing a 1 in this bit.

IdleIE = 1 enables interrupts when the controller sets the Idle flag. When software uses a DMA channel to provide data to the P1284 controller, it can be expected that the channel will do so in a timely manner, and thus, that an Idle condition signifies that the channel has finished transferring the block. (Software can also enable an interrupt from the DMA channel, but on the transmit side, such interrupts are not well-synchronized to events on the P1284 controller.) Conversely, if software provides data, Idle may not be grounds for an interrupt.

Some modes set the Idle flag when they are entered. However, such a setting of Idle never requests an interrupt.

StatIE = 1 enables "status" interrupts that are described separately for each mode.

DREQIE = 1 enables interrupts when the controller sets DREQ, except that in those modes that set DREQ when they are entered, such setting doesn't request an interrupt.

Table 3. Bidirectional Centronics Mode Selection

MODE	
0000	Non-P1284 mode
0001	Peripheral Compatible/Negotiation mode
0010	Peripheral Nibble mode
0011	Peripheral Byte mode
0100	Peripheral ECP Reverse mode
0101	Peripheral Inactive mode
0110	Peripheral ECP Forward mode with software RLE handling
0111	Peripheral ECP Forward mode with hardware RLE expansion
1000	Host Negotiation mode
1001	Host Compatible mode
1010	Host Nibble mode
1011	Host Byte mode
1100	Host ECP Forward mode
1101	Host Reserved mode
1110	Host ECP Reverse mode with software RLE handling
1111	Host ECP Reverse mode with hardware RLE expansion

Z80185 BIDIRECTIONAL CENTRONICS P1284 CONTROLLER (Continued)

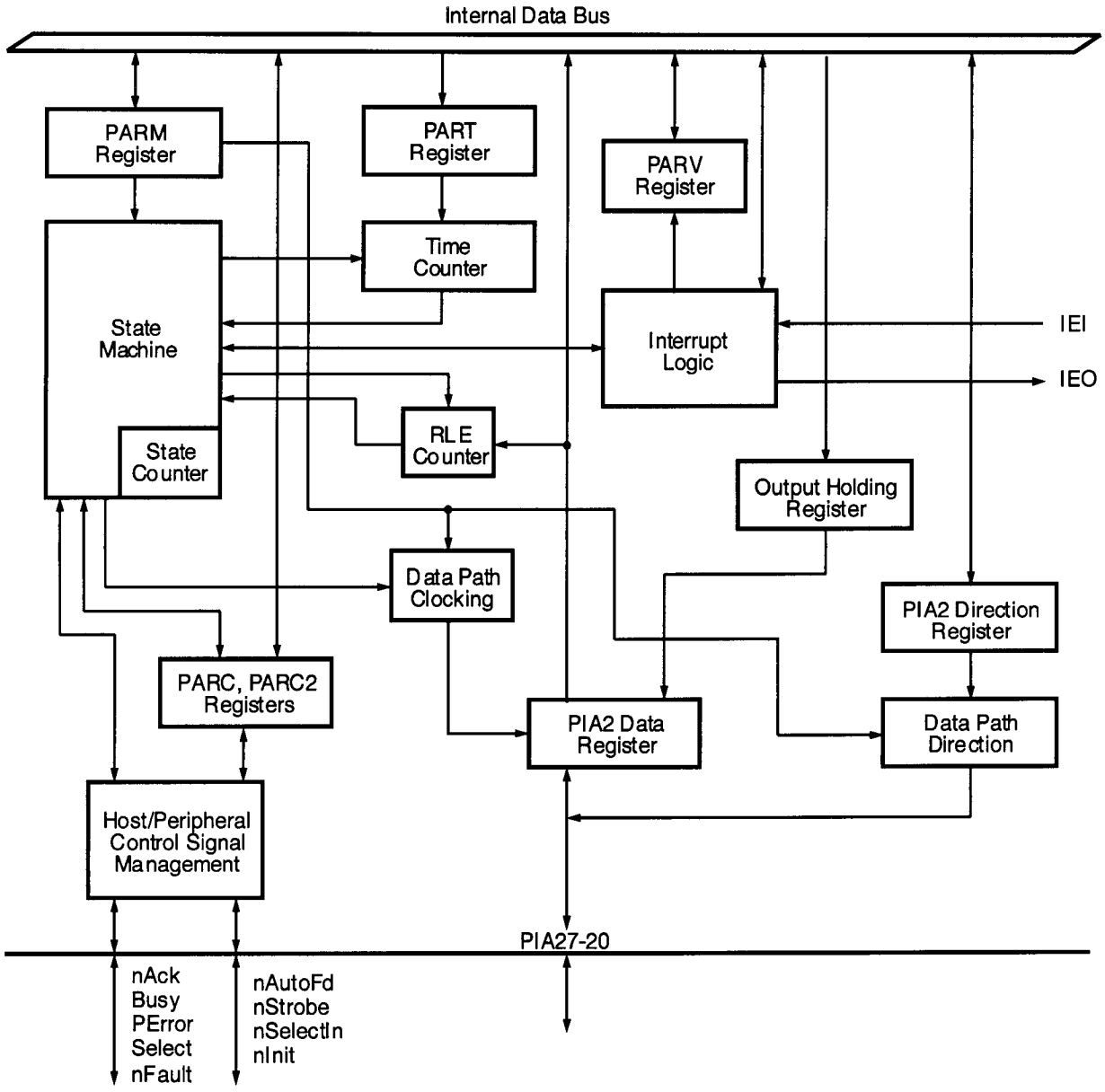


Figure 81. Bidirectional Centronics P1284 Controller Functional Block Description

Z80185 BIDIRECTIONAL CENTRONICS P1284 CONTROLLER (Continued)

Host Compatible Mode

1. Setting this mode configures PIA27-20 as outputs regardless of the contents of register E2. When entering this mode, the controller sets the Idle and DREQ bits, but these settings do not request an interrupt.

2. If software, or a DMA channel, writes eight bits to the Output Holding Register (OHR) when Idle is set, the controller transfers the byte to the Input/Output Register and negates DREQ only momentarily, so as to request another byte from software or the DMA channel.

3. In this mode, the nAutoFd line is not under control of the PARC register, but rather under control of which register the software uses to write data to the OHR. Each time the controller transfers a byte from the OHR to the Input/Output Register, it sets nAutoFd High if the byte was written to address E3, and Low if the byte was written to the "alternate" address EE. In a DMA application all of the bytes transferred from one output buffer will have the same state of nAutoFd, but this state can be changed from one buffer to the next by changing thel/O address used by the DMA channel. In non-DMA applications software can set the state of nAutoFd for each character, by writing data to the two different register addresses.

4. When a data byte has been valid on PIA27-20 for 750 ns (as controlled by the PART register), and the Busy and PError lines are Low and the Select, nAck, and nFault lines are High, the controller drives nStrobe Low. After the controller has held nStrobe Low for 750 ns it drives nStrobe back to High. Then it waits for 750 ns of data hold time to elapse. If software or a DMA channel has written another byte to the Output Holding Register (thus clearing DREQ) by the time this wait is satisfied, the controller transfers the byte from the Output Holding Register to the Input/Output Register, sets DREQ again, and returns to the event sequence at the start of this paragraph. Otherwise, it sets Idle and returns to the event sequence at the start of paragraph #2.

Status interrupts in this mode include rising and falling edges on PError, nFault, and Select.

Host Negotiation Mode

Setting this mode puts PIA27-20 under control of registers E2 and E3, as on the Z80181.

Software has complete control of the controller, and can either revert to Host Compatibility mode, or set one of the following Host modes, depending on how the peripheral responds to the Negotiation value(s).

Status interrupts in this mode include rising and falling edges on PtrClk (nAck), nAckReverse (PError), and nPeriphRequest (nFault). nFault is not used during actual P1284 negotiation, but is included because these events are significant during Byte and ECP mode idle times.

Host Reserved Mode

This mode differs from Host Negotiation mode only in that there are no status interrupts in this mode.

Peripheral Compatible/Negotiation Mode

In this mode, if P1284Active (nSelectIn) is Low, the controller sets PIA27-20 as inputs, regardless of the contents of register E2; when P1284Active (nSelectIn) is High, PIA27-20 are under the control of registers E2 and E3. On entry to this mode, the controller sets the Idle bit, if DREQ is set from a previous mode.

If, in this mode, nStrobe goes (is) Low, P1284Active (nSelectIn) is Low, and DREQ is 0, indicating that any previous data has been taken by the processor or DMA channel, the controller captures the data on PIA27-20 into the Input/Output Register, sets DREQ to notify software or the DMA channel to take the byte, drives the Busy line High, and one PHI clock later drives nAck Low. When at least 500 ns (as controlled by the PART register) have elapsed, the controller drives nAck back to High. One PHI clock later, if the CPU or DMA has taken the data and thus cleared DREQ, the controller drives Busy back to Low, otherwise it sets Idle.

Select, PError and nFault are under software control in this mode, and nAutoFd can be sensed by software, but has no other effect on operation.

Z80185 BIDIRECTIONAL CENTRONICS P1284 CONTROLLER (Continued)

Peripheral Nibble Mode

1. Software shouldn't set this mode until there is reverse data available to send. In other words, it should implement the P1284 "reverse idle mode" via software in Peripheral Compatibility/Negotiation mode. After software has driven nDataAvail (nFault), AckDataReq (PError), and Xflag (Select) all Low to signify that data is available, then driven PtrClk (nAck) High after 500 ns, and if requested programmed a DMA channel to provide data to send, when it sees HostBusy (nAutoFd) Low to request data, software should set this mode.

Setting this mode sets DREQ and Idle, but these settings do not request an interrupt. The PIA27-20 pins remain configured for data input but are not used. Instead, four of the five control outputs are driven with the LS and MS four bits of the Input/Output Register, as shown in Table 2, while PtrClk (nAck) serves as a handshake/clock output. On entering this mode the hardware begins routing bits 3-0 of the IOR to these lines.

2. If software, or a DMA channel, writes a byte to the Output Holding Register when Idle is set, the controller immediately transfers the byte to the IOR and clears Idle, and negates DREQ only momentarily to request another byte from software or the DMA channel.

3. After data has been valid on the four control outputs for 500 ns (as controlled by the PART register), the controller drives the PtrClk (nAck) line Low. Then it waits for the host to drive the HostBusy (nAutoFd) line back to High, after which it drives PtrClk (nAck) back to High, switches the four control lines to bits 7-4 of the IOR, and begins waiting for the host to drive HostBusy (nAutoFd) back to Low. When bits 7-4 have been valid for 500 ns and the host has driven HostBusy (nAutoFd) Low, the controller drives PtrClk (nAck) Low again and begins waiting for the host to drive HostBusy (nAutoFd) High. When HostBusy (nAutoFd) has been driven High, the controller returns the four control outputs to the state set by software in PARC. At this point, if software or a DMA channel has not yet written another byte to the Output Holding Register (thus clearing DREQ), the controller sets Idle and waits for software to do so. If/when software or a DMA channel has written a new byte to the OHR, the controller transfers the byte to the IOR, sets DREQ, and clears Idle if it had been set. Then, when the control outputs have been valid for 500 ns, the controller drives PtrClk (nAck) to High. It then waits for the host to drive HostBusy (nAutoFd) back to Low, at which time it switches the four control lines back to bits 3-0 of the IOR and returns to the event sequence at the start of this paragraph.

If there is no more data to send, when the controller sets Idle, software should modify PARC to make nDataAvail (nFault) and AckDataReq (PError) High, and then change the mode to Peripheral Compatible/Negotiation. Then (after 500 ns) software should set PtrClk (nAck) back to High in PARC and enter Reverse Idle state.

Status interrupts in Peripheral Nibble mode include rising and falling edges on P1284Active (nSelectIn) and nInit. The controller sets the IllOp (Illegal Operation) bit if P1284Active (nSelectIn) goes Low in this mode, before it drives nAck High for the status states on the four control lines, or after the host drives HostBusy Low thereafter, in which case software should immediately enter Peripheral Compatibility/Negotiation mode. If P1284Active goes Low, but IllOp stays 0, indicating that the Host negated P1284Active in a legitimate manner, software should enter Peripheral Inactive mode for the duration of the "return to Compatibility mode", and then enter Peripheral Compatibility/Negotiation mode.

Host Byte Mode

1. When in Host Negotiation mode the software has presented the value hex 01 or 05 on PIA27-20, it has been acknowledged by the peripheral, and the peripheral has driven nDataAvail (nFault) and AckDataReq (PError) to Low to indicate data availability and then driven PtrClk (nAck) back to High, software should set this mode. This sets PIA27-20 as inputs regardless of the contents of register E2, and clears the Idle flag. The controller then waits 500 ns (as controlled by the PART register) before proceeding.

2. For each byte, the controller drives HostBusy (nAutoFd) Low to indicate readiness for a byte from the peripheral. Then it waits for PtrClk (nAck) to go Low, at which time it captures the state of PIA27-20 into the Input/Output Register; sets the DREQ bit to request software, or the DMA channel to take the byte, and drives HostBusy (nAutoFd) High and HostClk (nStrobe) Low. When software, or the DMA channel, has taken the byte (thus clearing DREQ) and the peripheral has driven PtrClk (nAck) back High, and at least 500 ns after driving HostClk (nStrobe) Low, the controller drives HostClk (nStrobe) back to High, and samples nDataAvail (nFault). If it is still Low, the controller returns to the event sequence at the start of this paragraph, otherwise it sets the Idle flag.

In response to Idle, software should enter Host Negotiation mode. Thereafter, it can set HostBusy (nAutoFd) Low, to enter Reverse Idle state, or enter Host Compatible mode (reference IEEE P1284 specification), or conduct a new negotiation.

If software is programmed not to accept all the data that a peripheral has available in this mode, it should first disable the DMA channel, if one is in use, and then wait for DREQ to be 1 and nAck to be 1. Then it should reprogram the controller back to Host Negotiation mode, read the last byte from the IOR, drive HostClk (nStrobe) back to High, and then drive P1284Active (nSelectIn) Low to instruct the peripheral to leave Byte mode.

There are no status interrupts in Host Byte mode.

Peripheral Byte Mode

1. Software should not set this mode until there is reverse data available to send — that is, it should implement the P1284 "reverse idle mode" via software in Peripheral Compatibility/Negotiation mode. The exact sequencing among PtrClk (nAck), nDataAvail (nFault), and AckDataReq (PError) differs according to whether this mode is entered directly from Negotiation or from reverse idle phase, and is controlled by software. But in either case, before software sets this mode, it should set nDataAvail (nFault) and AckDataReq (PError) to Low, then after 500 ns, set PtrClk (nAck) to High. When it detects that the host has driven HostBusy (nAutoFd) Low to request data, software should set this mode, which sets the DREQ and Idle flags.

2. In this mode, as long as P1284Active (nSelectIn) remains High, the controller drives PIA27-20 as outputs, regardless of the contents of register E2. When software, or a DMA channel, writes the first byte to the Output Holding Register, the controller immediately transfers the byte to the Input/Output Register, clears Idle but negates DREQ only momentarily, to request another byte from software, or the DMA channel.

3. After each byte is transferred to the IOR, the controller waits 500 ns data setup time (as controlled by the PART register) before driving PtrClk (nAck) Low, and thereafter waits for the host to drive HostBusy (nAutoFd) High. When this occurs, if software, or the DMA channel, has not written more data to the Output Holding Register, that is, if DREQ is still set, the controller sets the Idle flag and waits for software or the DMA channel to do so. If software, or the DMA channel, then writes data to the Output Holding Register, the controller clears DREQ and Idle. When there is data in the OHR and DREQ is 0, this guarantees that it is appropriate to keep nDataAvail (nFault), and AckDataReq (PError) Low to indicate that more data is available, and the controller drives PtrClk (nAck) back to High. The controller then waits for a rising edge on HostClk (nStrobe), and then for the host to drive HostBusy (nAutoFd) Low, at which time it transfers the byte from the OHR to the Output Register, sets DREQ, and then it returns to the event sequence at the start of this paragraph.

While this mode is in effect, software should monitor the interface for two conditions:

Case 1: Idle set and no more data to send, or

Case 2: P1284Active (nSelectIn) Low.

In Case #1, the software should write zero to register E3 to keep PIA27-20 outputs momentarily, and then set the mode back to Peripheral Compatibility, so that the interface is fully under software control, set nDataAvail (nFault) and AckDataReq (PError) High to signify no more data, wait 500 ns, and set PtrClk (nAck) back to High. When HostBusy goes back to Low, the software should set PIA27-20 back to inputs.

In Case #2, if a falling edge on P1284Active happens any time other than between a rising edge on HostClk (nStrobe), and the next falling edge on HostBusy (nAutoFd), the controller sets the IllOp bit to notify software that an immediate Abort is in order, in which case software should immediately enter Peripheral Compatibility/Negotiation Mode. If P1284Active goes Low, but IllOp is not set, meaning that the Host negated P1284Active in a "legal" manner, software should enter Peripheral Inactive Mode for the duration of the "return to Compatibility Mode", and then enter Peripheral Compatibility/Negotiation Mode.

Status interrupts in Peripheral Byte Mode include rising and falling edges on P1284Active (nSelectIn) and nInit.

System Configuration Register

This register controls a number of device-level features on the Z80185 and includes the following control bits:

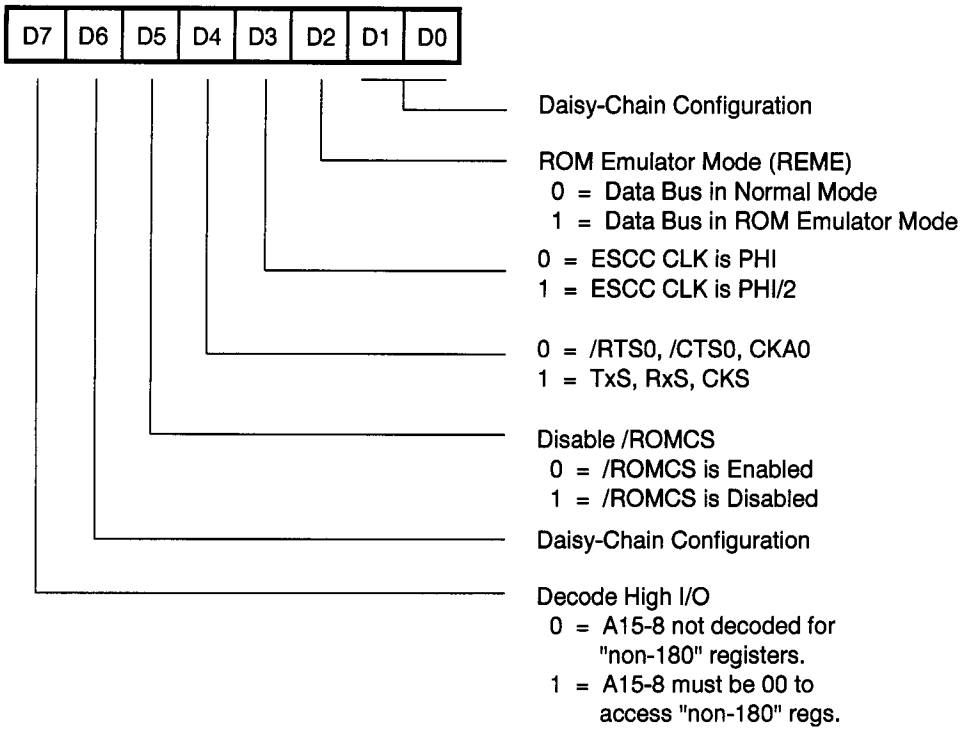


Figure 82. System Configuration Register
(I/O Address %ED)

Z80185 CTC, AND MISCELLANEOUS REGISTERS (Continued)

RAM And ROM Registers

Three registers, ROMBR, RAMLBR and RAMUBR, and two pins, /ROMCS and /RAMCS, assist with decoding of ROM and RAM blocks of memory.

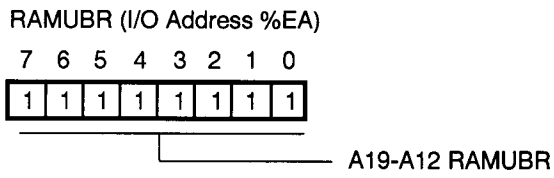


Figure 83. RAMUBR (I/O Address %EA)

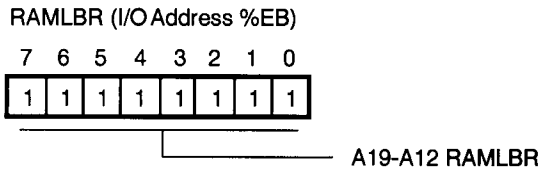


Figure 84. RAMLBR (I/O Address %EB)

The names RAMUBR and RAMLBR stand for RAM Upper Boundary Range and RAM Lower Boundary Range. These two registers specify the address range for the /RAMCS signal. When accessed, memory addresses are less than, or equal, to the value in the RAMUBR, and greater than, or equal to, the value programmed in the RAMLBR, /RAMCS is asserted.

ROMBR ROM Address Boundary Register

This register specifies the address range for the /ROMCS signal. When an accessed memory address is less than, or equal to, the value programmed in this register, but greater than the size of on-chip ROM (if on-chip ROM is enabled), the /ROMCS signal is asserted.

/ROMCS can be forced to a "1" (inactive state) by setting bit 5 in the System Configuration Register, to allow the user to overlay the RAM area over the ROM area.

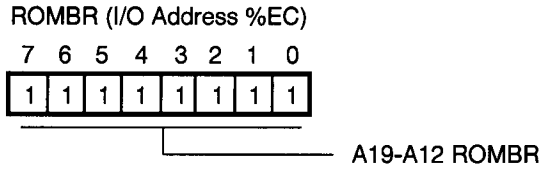


Figure 85. ROMBR (I/O Address %EC)

/RAMCS and /ROMCS are active for accesses by an external master, as well as by the Z80185 processor. If/ROMCS and /RAMCS are programmed to overlap,/ROMCS is asserted and /RAMCS is inactive for addresses in the overlapping region.

Chip Select signals are active for the address range:

/ROMCS: (ROMBR) >= A19-A12 >=
Size of On-Chip ROM (if enabled, else 0)

/RAMCS: (RAMUBR) >= A19-A12 >= (RAMLBR)

All three of the above registers are set to "FFh" at Power-On Reset. This means that if on-chip ROM is enabled,/ROMCS is asserted for all addresses above the size of on-chip ROM, if not, /ROMCS is asserted for all addresses. Since /ROMCS takes priority over /RAMCS, the latter will never be asserted until the value in the ROMBR and RAMLBR registers are re-initialized to lower values.